ITCS 209 Object Oriented Programming	Name:	**Lab**	**Challenge Bonus**	**Peer Bonus**
	ID:			
	Sec:			

Lab06: Designing Class

You have been hired to implement a system for Ski Jumping Sport in the Winter Olympic 2022. For the brief overview, many competitors will have to compete by achieving the farthest jump after sliding down on their skis from a specially designed curved ramp. To simplify this, the score of each competitor comes from two factors that are (1) jump length and (2) the competitor's aerial style which is scored by 6 panels of judges. Our system will keep track of scores and aggregate them to find the best 3 who get a competition's gold medal. The detail of the requirement are as follows:

The program must have **Three** following classes (from the starter pack):

- The `Score` class stores scores and provide calibrated average about the scores from 6 judges. This will be used as a part of the `Competitor` class.
- The `Competitor` class represent a competitor in the competition. This class will store attributes related to the competitor including the competitor's scores from jumping. This class will be used in the `Competition` class.
- The `Competition` class represents a round in competition that involves multiple competitors.

Task 1: Implement the `Score` Class

The class `Score` is mainly used to keep two scores (1) the jump length and (2) the aerial style given by 6 judges. Your task is to complete all methods, including the constructor in `Score.java`. As the score for the aerial style (2) was given by 6 judges, the score needs to calibrate by excluding the minimum and the maximum score from 2 judges before calculating the average.

Expected Output (Score.java):

```
9.5
[5.5, 4.5, 3.5, 2.5, 1.5, 6.5]
4.0
```

Task 2: Implement the `Competitor` Class

The class `Competitor` keeps the competitors' scores and profiles (*FirstName, LastName, Age, Nationality*). Your task is to complete all methods, including the constructor in the `Competitor.java`. As the jump length (1) cannot be directly used as a jump score (jScore) , your need to scale it to the range [0.0, 10.0] by using the following formula:

```
jScore = (jump length / maxKpoint) *10
```

`maxKpoint` is a construction point used to calculate the score, in this task the `maxKpoint` will be fixed as a constant to 50 meters

As each competitor can perform more than 1 round, the method `showRawScore(int round)` will display all raw scores of the competitor from the given round as expected output below:

Expected Output for **jump length=45** and **aerial style = {8.0,8.5,9.0,7.5,6.7,8.0}**.

```
Jumpping score: 9.0
Ariel Style (Raw) score: |8.0|8.5|9.0|7.5|6.7|8.0|
Final score: 8.50
```

The method `getFinalScore(int round)` will compute and return the final score by the following formula:

```
Final score = jScore*w1 + sScore*w2
```

`w1` and `w2` are the weight of the score range from [0.0, 1.0], this must be defined since calling the constructor `Competitor()`. The `jScore` comes from method `getJumpScores()` in the class `Score`. The `sScore` can obtain from method `getCalibratedAverage()` in the class `Score`. The expected output when complete the whole method is as below:

Expected Output (Competitor.java)

```
===============================
Name: Somechai Jaidee, Age:33, Nationality:Thai
[Round1]
Jumpping score: 9.0
Ariel Style (Raw) score: |8.0|8.5|9.0|7.5|6.7|8.0|
Final score: 8.50
[Round2]
Jumpping score: 9.8
Ariel Style (Raw) score: |10.0|8.5|9.0|7.5|6.7|8.0|
Final score: 9.03
===============================
Name: Brain Toss, Age:35, Nationality:British
[Round1]
Jumpping score: 7.4
Ariel Style (Raw) score: |9.0|9.5|9.0|7.5|8.7|8.0|
Final score: 8.42
[Round2]
Jumpping score: 9.6
Ariel Style (Raw) score: |9.3|8.5|9.5|8.5|7.7|8.0|
Final score: 8.78
```

Note that:

- You are **not allowed** to (1) modify any given attributes or method, (2) use the predefined method to max, min, and method to sort the element.
- You are free to create any helper class as required.
- The main class are given to all files for testing.
- More detail will be given in the **starter pack**.
- Good Luck!!

Challenge Bonus (Optional):

Task 3: Implement the `SkiCompetition` Class

The `SkiCompetition` represent a competition that involves multiple competitors. This class maintain a list of competitors and their scores. You have to implement the method `getGoldMedal(List<Competitor> competitors, int round)`, which take a list of all competitors and round, then return a list of **top 3 highest (final) score competitors** of the given round. There is also an important condition to get a gold medal if the final score is lower than 8.0, the competitor must be excluded from the list.

Expected Output (SkiCompetition.java)

```
[Round1]
== Scores Board ==
Name: Brian Xavia, Age:33, Nationality:American | Final Score: 8.94
-----------------
Name: Albert Wattana, Age:44, Nationality:French | Final Score: 8.28
-----------------
Name: Maja Luvinia, Age:34, Nationality:Polish | Final Score: 9.28
-----------------
Name: Joby Beckett, Age:21, Nationality:Australia | Final Score: 9.02
-----------------

== List of Gold Medal ==
Name: Brian Xavia, Age:33, Nationality:American | Final Score: 8.94
-----------------
Name: Maja Luvinia, Age:34, Nationality:Polish | Final Score: 9.28
-----------------
Name: Joby Beckett, Age:21, Nationality:Australia | Final Score: 9.02
-----------------

=================================
[Round2]
== Scores Board ==
Name: Brian Xavia, Age:33, Nationality:American | Final Score: 5.80
-----------------
Name: Albert Wattana, Age:44, Nationality:French | Final Score: 6.26
-----------------
Name: Maja Luvinia, Age:34, Nationality:Polish | Final Score: 6.31
-----------------
Name: Joby Beckett, Age:21, Nationality:Australia | Final Score: 8.07
-----------------

== List of Gold Medal ==
Name: Joby Beckett, Age:21, Nationality:Australia | Final Score: 8.07
-----------------
```